applied sciences

MDPI

Article

An IP Multimedia Subsystem Services Proxy Gateway Based on a JAVA Dynamic Module System

Yi-Chun Chang [1], Jian-Wei Li [2,*] and Jing-Hong Lv [2]

[1] Department of Computer Science and Information Engineering, Hungkuang University, Taichung City 43302, Taiwan; changyc@hk.edu.tw

[2] Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung City 41349, Taiwan; s10030042@gmail.com

* Correspondence: lijw@cyut.edu.tw

Received: 30 September 2018; Accepted: 21 October 2018; Published: 25 October 2018

Abstract: By virtue of the rapid development of the Information and Communication Technology (ICT), people have owned various devices connected to the Internet to enjoy various emerging application services, such as multimedia related services. IP Multimedia Subsystem (IMS) is the integration platform for multimedia application services and provides a rich feature set of converged services in the Next Generation Network (NGN). IMS services are accessed by a user anytime and anywhere through his/her personal self IMS device, e.g., mobile device, equipped with a Universal Subscriber Identity Module (USIM) card. People have owned various devices connected to the Internet at homes or offices. However, some devices can not access IMS services without USIM cards embedded, which is called non-IMS devices. People wish that more devices without USIM cards embedded should be connected to the Internet on which IMS services are accessed in addition to the mobile devices equipped with USIM cards. Accordingly, we propose an Open Service Gateway Initiative (OSGi)-based IMS service gateway, in which OSGi technology is the dynamic module system for JAVA. The proposed gateway is authorized by a personal self IMS device such that IMS services are accessed by non-IMS devices, in this paper. Moreover, these non-IMS devices rely on different networks to access technologies and network protocols and the IMS service gateway proposed here adopts the OSGi framework for no problem of integration.

Keywords: bundle; integration; IP Multimedia Subsystem; OSGi framework; proxy

1. Introduction

In the IoT (Internet of Things) era of information and communications technologies (ICT) booming and maturing, a user may own many peripheral devices in addition to the owned self-mobile device (e.g., smartphone). Meanwhile, many emerging application services have flourished, such as IP Multimedia Subsystem (IMS) related application services.

Cellphone users in the 2G era, who inserted a Subscriber Identity Module (SIM) card into a traditional mobile phone for communications services, opt to insert a Universal Subscriber Identity Module (USIM) card into a smartphone and enjoy the IMS based Voice over IP (VoIP) service in the 4G era or later. IMS, which is one of the critical cores of the next-generation network (NGN), is a platform on which multimedia application services are integrated for more IMS-based services in addition to VoIP emerging in the future [1–5].

People wish to enjoy the same application services on these peripheral devices and the owned self-mobile device in their home or office. However, IMS services are accessed restrictively on a personal self IMS device in which the USIM card is inserted only. Therefore, these peripheral devices in their home or office can not access IMS services, in which such devices are called non-IMS devices. For example, the device for dialing or answering a Session Initiation Protocol (SIP) call during video/voice communications is the user's personal self IMS device only in which the USIM card is embedded rather than any communications device moved or preferred by a user. To overcome this problem, we present the IMS service proxy gateway with which a user in a trusty field domain; for example, home, enjoys IMS services such as communication by the IMS device at peripheral devices (non-IMS devices) instead of the personal personal self IMS device only.

OSGi [6] technology, which is the dynamic module system for JAVA, was used in the development of the residential gateway in past studies frequently for the members of the home network to access multimedia. OSGi technology is a Java-based technical platform but lots of software components for processing image transmission are not activated with Java. Most works of literature employ the OSGi technology to deploy a gateway for smart home application or Internet of things (IoT), vehicular network and so no. [7–13]. Fewer works of literature employ it to handle the rich applications, such as multimedia. For example, Dr. Lai et al. integrated DLNA (Digital Living Network Alliance) services into the OSGi framework in the extension program of multimedia video sharing on the DLNA local network through which the home media resources were accessed by users via the peer-to-peer (P2P) network [14,15]. However, the applications with respect to video phone calls or even IMS services are still uninvolved despite lots of research for the management of multimedia services based on the OSGi framework. In 2018, Andrade et al. [16,17] proposed the application server (AS) for IMS, named SANGN, which is based on the OSGi technology with Java Agent DEvelopment (JADE) framework. Nevertheless, the AS is only for an IMS device, not for a non-IMS device. Accordingly, to settle the problem of IMS services not available to peripheral devices (non-IMS devices) in a field domain, we employ the OSGi framework to deploy the proposed IMS service proxy gateway with which the users inside a trusty field domain, e.g., home, make use of peripheral devices in the field domain to enjoy IMS services, for example, communication session.

2. Background

2.1. IP Multimedia Subsystem (IMS)

The IMS Network, i.e., IP Multimedia Subsystem Network, which is the control layer of the Next Generation Network (NGN) and was formulated by the 3GPP (3rd Generation Partnership Project) and 3GPP2 [1–4], is used to access different network technologies such as Wi-Fi, LTE, 3G/3.5G and wired network and depends on SIP (Session Initiation Protocol) [18] as the communications protocol to control SIP signals by the Call Session Control Function (CSCF). The brief network architecture of IMS is shown in Figure 1.

- P-CSCF (Proxy-CSCF): As the first IMS network server faced by one user, P-CSCF forwards a user's messages into IMS, SIP messages to I-CSCF for further processing, and any SIP signal response to a user.
- I-CSCF (Interrogating-CSCF): I-CSCF is a query server that relies on a user's SIP requests or information brought by P-CSCF to find an appropriate S-CSCF for relevant services and saves the information in Home Subscriber Server (HSS) for other queries later.
- S-CSCF (Serving-CSCF): S-CSCF is a service server which is particularly used in processing a user's service requests and depends on the requests to find a corresponding application server.
- HSS (Home Subscriber Server): HSS in IMS is particularly used in storing users' data such as e-mail, phone number and user's authority to access services.
- Application Server (AS): As a top-layer application server such as a multimedia streaming and monitoring system, AS depends on S-CSCF to give user services.

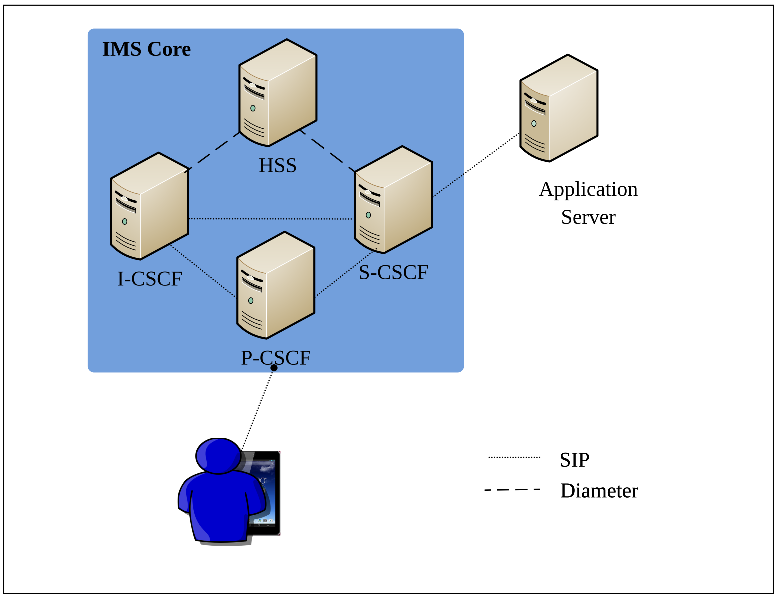


Figure 1. Brief network architecture of IMS.

2.2. OSGi (Open Service Gateway Initiative)

With the capacity of "integration of software and hardware", the OSGi (Open Service Gateway Initiative) platform is characteristic of heterogeneous technologies integrated into the same platform and interacting with one another [6]. Currently, the OSGi platform is an open service platform for application programs and remote management of network appliances including mobile phones activated in indoor environments, vehicles or others. In the software framework available in OSGi, the various software services are transformed to different service modules or so-called Bundles that are carried in the OSGi framework through OSGi core services and installed, enabled, upgraded or unloaded remotely without rebooting according to the management architecture of Bundles and the flowchart for the life cycle of a Bundle [6].

3. System Architecture

As Figure 2 depicted, a user is in his trusted domain, such as home or office. The user owns his personal self IMS device and other peripheral devices (non-IMS devices). In addition, an OSGi-based IMS service proxy gateway resides in this trusted domain. The user can share IMS service with the IMS service proxy gateway; then, other non-IMS devices on this domain can access IMS services with the help of the IMS service proxy gateway.

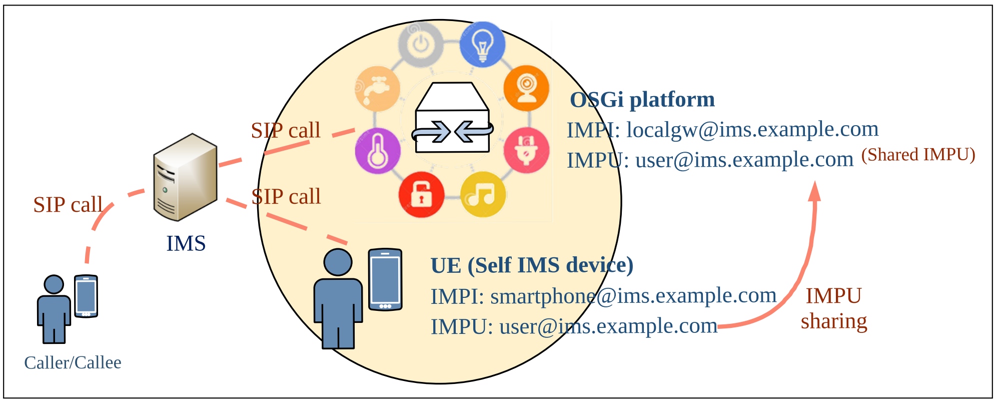

OSGi platform
IMPI: localgw@ims.example.com
IMPU: user@ims.example.com (Shared IMPU)
SIP call
SIP call
SIP call
IMS
Caller/Callee
UE (Self IMS device)
IMPI: smartphone@ims.example.com
IMPU: user@ims.example.com
IMPU sharing

Figure 2. System architecture.

3.1. IMS Service Proxy Gateway

In the IMS service proxy gateway, OSGi which plays the role of a middleware is used to integrate software and hardware services and effectuate the IMS service proxy gateway. As shown in Figure 3, the gateway includes: (1) OSGi framework; (2) functional modules (Bundles); and (3) configuration services. With Bundles carried in the OSGi framework via OSGi core services, all functional modules in which interdependency is created can be upgraded and unloaded conveniently. Moreover, the Bundles can be activated and configured flexibly by compiling the configuration service.

The function/mechanism supported by the IMS service proxy gateway is available to a respective Bundle, for example, the IMS Bundle and the Real-time Transport Protocol (RTP) Bundle are used to manage protocols with respect to IMS services such that IMS services are effectuated by the IMS service proxy gateway. In addition to the above functional Bundles, other types of Bundles are also designed for basic operations.

- *Core Bundle*: The Core Bundle provides a mechanism to manage events for interconnectivities between different types of software/hardware services and the OSGi framework via OSGi core services.
- *I/O Bundle*: The I/O Bundle takes charge of inputs/transfers of network services.
- *Model Bundle*: The Model Bundle is responsible for configuring parameters between all types of modules for the IMS service proxy gateway such that the flexible operation modes between objects via software configurations are available.
- *User Interface (UI) Bundle*: The UI Bundle provides a user interface through which the functional module, REST (Representational State Transfer) Bundle [19], for resources of the IMS service proxy gateway is accessed.
- *Action Bundle*: The Action Bundle is characteristic of additionally incorporating a distinct automated controllable context which is activated with the Action Bundle matching the configuration service.

The IMS service proxy gateway has to support processing of IMS services or is a gateway to integrate peripheral software/hardware services in a field domain through OSGi in case of no capacity to process IMS services. Accordingly, the service modules such as IMS Bundle, RTP Bundle and Session Mobility Bundle should be incorporated in the IMS service proxy gateway, as shown in Figure 4. With the core protocol of SIP, the IMS Bundle is applicable to processing exchanged SIP signals specifically such that the true media transfer is handled by the RTP Bundle after completion of connectivity.

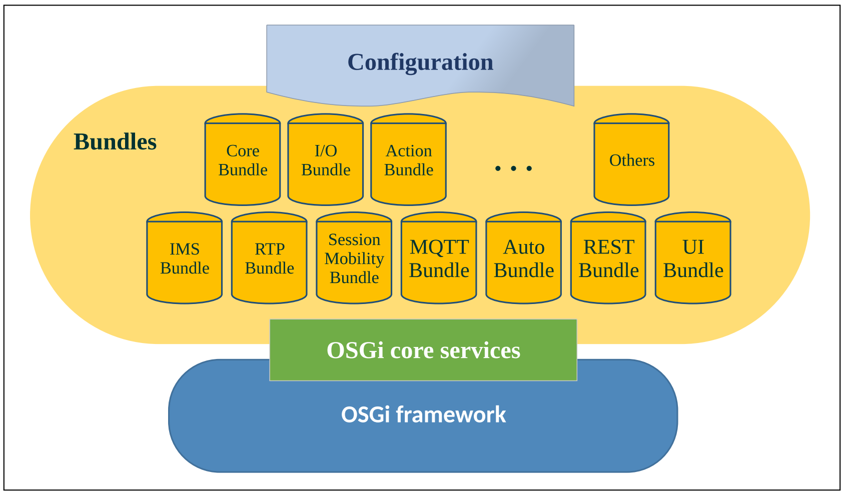


Figure 3. Core architecture of the IMS service proxy gateway.

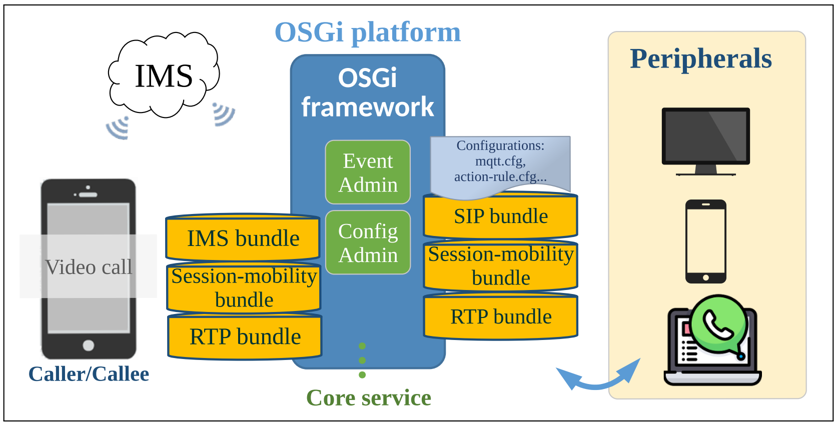


Figure 4. Architecture for operations between IMS and relative Bundles.

3.2. The Involved Bundles between UE and Peripheral Devices

In the IoT era, more and more peripheral devices are carried by users moving in various field domains. For applications in different circumstances, the types of software/hardware services for IoT are different from one another and provided with respective User Interfaces (UIs) automatically which are fundamental to applications in multiple field domains. Because of the characteristic of an OSGi gateway, the individual facilities are not configured on UIs in advance and the members in a field domain are controlled dynamically. As shown in Figure 5, the REST (Representational State Transfer) service [19] adopted in this paper and taken as the medium to access shared information on the "IMS service proxy gateway" by personal self IMS devices is enabled through hyperlinks without interventions of other discovery mechanisms on the personal self IMS devices. As such, the extendibility of a gateway is significantly improved in virtue of no connectivity required. Figure 5 illustrates the resources shared on the OSGi platform are accessed by user equipment (UE) through the REST service and information is received and displayed on UI dynamically after analysis of any Extensible Markup Language (XML) responded.

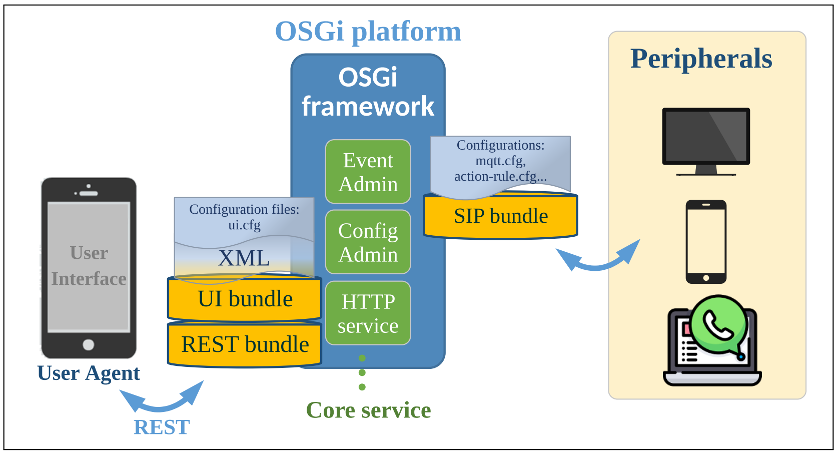


Figure 5. Information shared to peripheral devices on the OSGi platform and accessed by UEs through the REST service.

4. The IMS Service Sharing Mechanism

In the proposed IMS service proxy gateway, IMS related Bundles, such as IMS Bundle, RTP Bundle, and Session Mobility Bundle, have been added to the OSGI framework to handle IMS services. To realize the function, IMS service proxy, the IMS service sharing mechanism should be provided, in which contain both (1) IMPU sharing service and (2) the session establishment procedure in proxy mode.

4.1. IMPU Sharing and Available to Multiple Devices

The IMS user identifier consists of IMPI (IMS Private User Identity) and IMPU (IMS Public User Identity): IMPI which is a user's private ID for identity authentication corresponds to multiple IMPUs probably and has the format of usename@reaml [1–4]. Moreover, the identical IMPU can be shared by different IMPIs, as shown in Figure 2. According to these features, the IMPU of a user's personal self IMS device in this paper is shared to the IMS service proxy gateway which will be authorized to access IMS services owned by the user and answer an SIP call from a caller, as shown in Figure 6a. However, IMPI or IMPU, each of which corresponds to the other and exists independently for sharing, are limited to the identity of a single user rather than multiple ones simultaneously when multiple users and IMS devices probably exist in the field domain at the same time. Accordingly, the solution of IMPU sharing for multiple IMS services activated in the IMS service proxy gateway simultaneously is crucial when SIP calls are not answered directly in the field domain in which peripheral devices belong to another private individual. As shown in Figure 6b for the relationship between IMPI and Shared IMPUs, the IMS service proxy gateway is authorized to control multiple users' IMPUs at the same time for independent operations of IMS services to be handled.

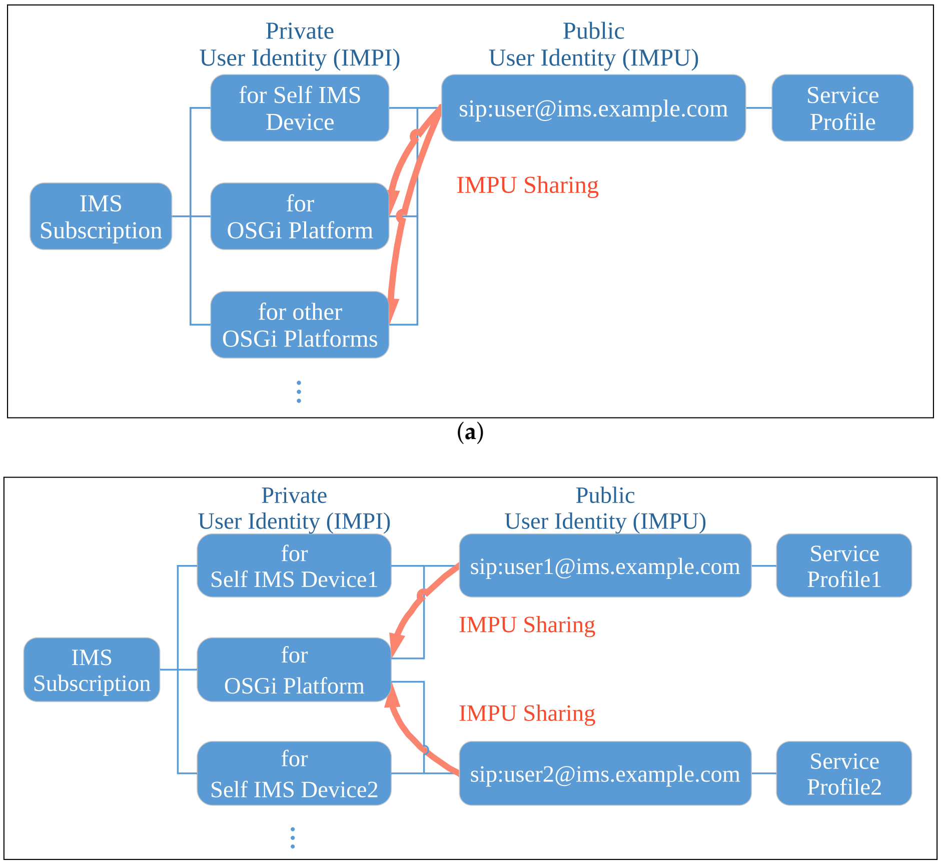


Figure 6. The relationship between IMPI and Shared IMPUs for (**a**) a personal self IMS device and the proxy gateway; and (**b**) multiple personal self IMS devices and the proxy gateway.

4.2. The Session Establishment Procedure in Proxy Mode

The IMS service sharing based on IMPU Sharing has been presented in previous sections. However, the same identity to be shared by multiples devices means a SIP account requested and responded by several devices simultaneously, that is, the multiple responses of so-called SIP Forking, such that these devices are not recognized easily. Accordingly, Caller preference/Callee capabilities [20,21] and Q-value [21] are adopted for no multiple responses attributed to SIP Forking.

- **Q-value solution** Q-value, which is a parameter of a SIP packet header, means UE's relative preference in a SIP request and the high Q-value has the high priority (preferential response to a SIP request). For Q-value, the Serial forking solution is applicable to the condition that the response preference of a Callee is unfamiliar to a Caller. Moreover, the Serial forking is a mechanism to decide the priority ordering for a destination according to the Q-value labeled on UE during the SIP registration (as specified in RFC 3841 [21]). In virtue of this solution, the response conflicts will not exist between personal self IMS devices and peripheral devices despite various networking facilities in a field domain.
- **Caller preference/Callee capabilities solution**
  This solution is characteristic of the Caller preference, e.g., mobile device or household appliance, configured in a session invitation such that the message routing of a SIP session is available to UEs with corresponding Callee capabilities rather than all devices. For that matter, the common description language to configure Caller preference/Callee capabilities [20,21] in SIP standard specifications is Media feature tag (as specified in RFC 5688 [22]) which is one condition of SIP Forking routing.
  For a Caller with the definite request preference, a session invitation can be initiated by the Caller through configurations of Caller preference/Callee capabilities. With the Media feature tag installed in an IMS device in advance and recorded in IMS S-CSCF during a registration step of the IMS device, the mechanism is simpler and more effective than the common technique of multicasting followed by filtering because the routing for a session request is available in the SIP proxy (IMS) without extensive broadcasting of request signals and the SIP signals are routed to a correct SIP UA. In the case of more than one device with same Callee capabilities during routing, the Q-value solution for recognition of the priority is further applicable for no conflict of two solutions.

Based on IMPU Sharing for sharing of IMS services, the IMS service proxy gateway in this paper presents the capacity of IMS service proxy through multiple responses of SIP Forking in the above two solutions. As shown in Figure 7, the IMS service for session sharing is created among peripheral devices (non-IMS devices) through the IMS service proxy gateway according to the following steps.

Step 1 The initial step for the gateway in a field domain available to a user includes (1) registration on the OSGi-based IMS service epoxy platform and (2) registration of the user's personal self IMS device.

Step 2 With IMS service proxy in Step 1 and Q-value configured in the gateway, the service proxy (that is, the gateway) in the field domain is the first Caller in an invitation of a phone session. Moreover, the handshaking of messages between the gateway and UEs depends on the information access mechanism of REST with three available conditions such as Confirmation, Accepted and ACK (Acknowledgment) signals.

Step 3 Furthermore, the peripheral devices to be incorporated in a phone session can be chosen manually or automatically according to users' preferences. In the manual mode, UEs choose the objects requesting Session passing through the REST protocol [19].

Step 4 The Session passing step is enabled by the gateway with the capacity of session proxy for sharing of media resources to peripheral devices around an object.

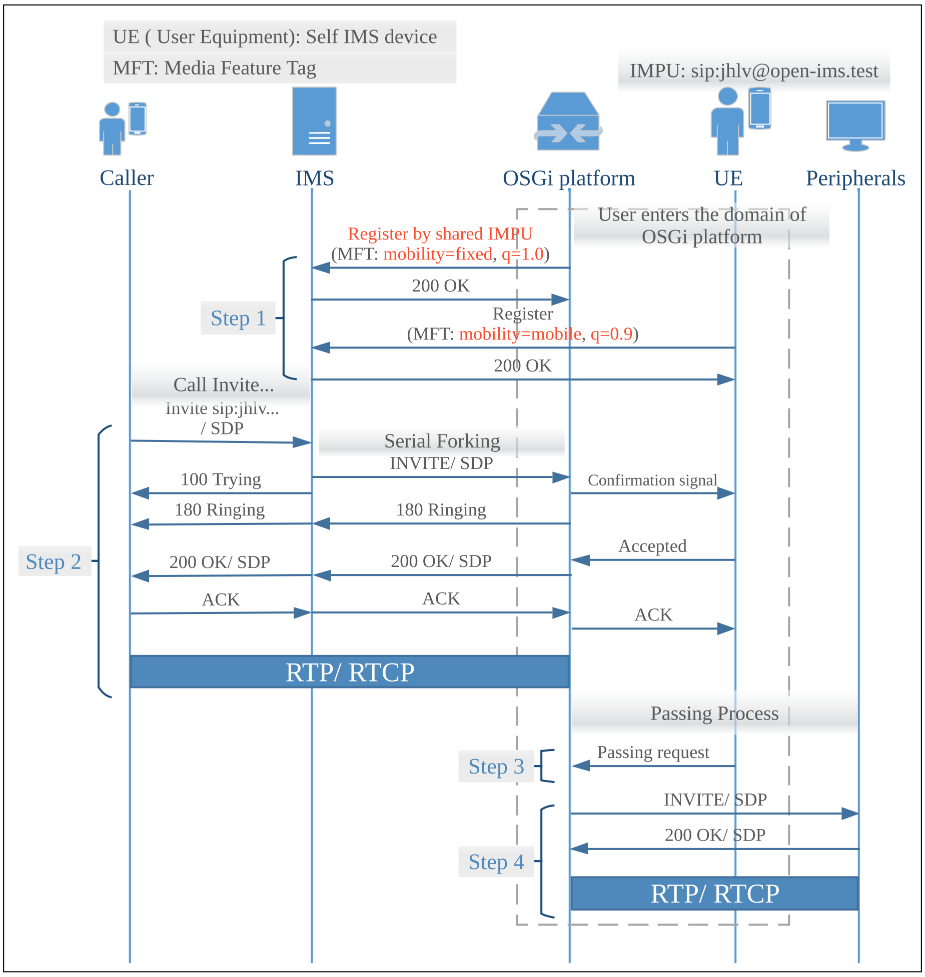


Figure 7. The session establishment procedure in the proxy mode.

5. Implementation

As shown in Figure 8, the objects incorporated in implementations of this paper consist of the IMS platform, the IMS service proxy gateway, personal self IMS devices, and peripheral devices (non-IMS devices).

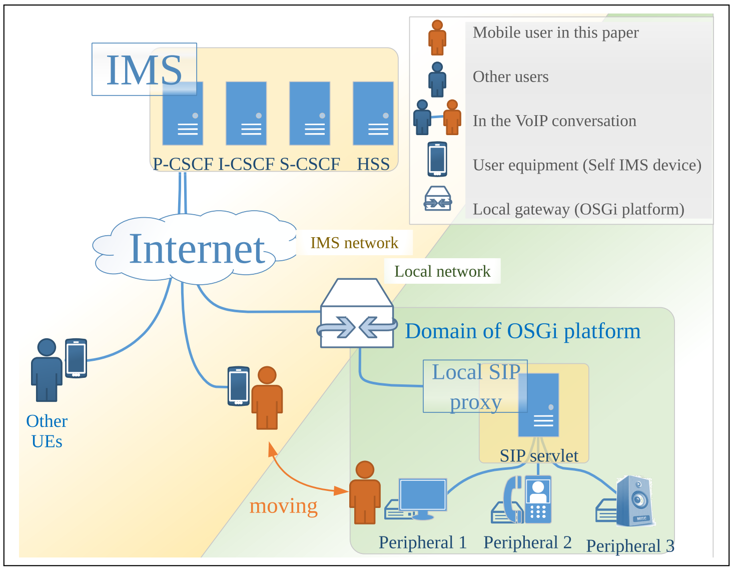
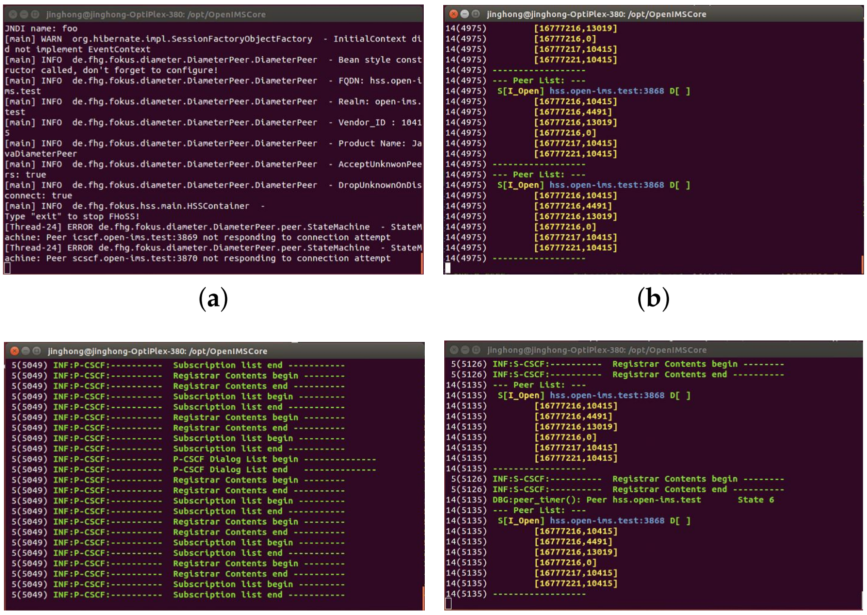


Figure 8. Architecture of the experimental environment.

The IMS network includes three CSCFs (P-CSCF, I-CSCF and S-CSCF), one HSS and one Application Server, which were set up based on OpenIMSCore [9] for development of the IMS network, as depicted in Figure 9. Moreover, the IMS service proxy gateway was established according to Eclipse Equinox [23] as the core framework; the IMS bundle and the RTP bundle were developed with JAIN-SIP [24] and Java Media Framework (JMF) [25], respectively.



jinghong@jinghong-OptiPlex-380: /opt/OpenIMSCore

```
JNDI name: foo
[main] WARN  org.hibernate.impl.SessionFactoryObjectFactory  - InitialContext di
d not implement EventContext
[main] INFO  de.fhg.fokus.diameter.DiameterPeer.DiameterPeer  - Bean style const
ructor called, don't forget to configure!
[main] INFO  de.fhg.fokus.diameter.DiameterPeer.DiameterPeer  - FQDN: hss.open-i
ms.test
[main] INFO  de.fhg.fokus.diameter.DiameterPeer.DiameterPeer  - Realm: open-ims.
test
[main] INFO  de.fhg.fokus.diameter.DiameterPeer.DiameterPeer  - Vendor_ID : 1041
5
[main] INFO  de.fhg.fokus.diameter.DiameterPeer.DiameterPeer  - Product Name: Ja
vaDiameterPeer
[main] INFO  de.fhg.fokus.diameter.DiameterPeer.DiameterPeer  - AcceptUnknwonPee
rs: true
[main] INFO  de.fhg.fokus.diameter.DiameterPeer.DiameterPeer  - DropUnknownOnDis
connect: true
[main] INFO  de.fhg.fokus.hss.main.HSSContainer  -
Type "exit" to stop FHoSS!
[Thread-24] ERROR de.fhg.fokus.diameter.DiameterPeer.peer.StateMachine  - StateM
achine: Peer icscf.open-ims.test:3869 not responding to connection attempt
[Thread-24] ERROR de.fhg.fokus.diameter.DiameterPeer.peer.StateMachine  - StateM
achine: Peer scscf.open-ims.test:3870 not responding to connection attempt
```

(**a**)

jinghong@jinghong-OptiPlex-380: /opt/OpenIMSCore

```
14(4975)         [16777216,13019]
14(4975)         [16777216,0]
14(4975)         [16777217,10415]
14(4975)         [16777221,10415]
14(4975) ------------------
14(4975) --- Peer List: ---
14(4975)  S[I_Open] hss.open-ims.test:3868 D[ ]
14(4975)         [16777216,10415]
14(4975)         [16777216,4491]
14(4975)         [16777216,13019]
14(4975)         [16777216,0]
14(4975)         [16777217,10415]
14(4975)         [16777221,10415]
14(4975) ------------------
14(4975) --- Peer List: ---
14(4975)  S[I_Open] hss.open-ims.test:3868 D[ ]
14(4975)         [16777216,10415]
14(4975)         [16777216,4491]
14(4975)         [16777216,13019]
14(4975)         [16777216,0]
14(4975)         [16777217,10415]
14(4975)         [16777221,10415]
14(4975) ------------------
```

(**b**)

jinghong@jinghong-OptiPlex-380: /opt/OpenIMSCore

```
 5(5049) INF:P-CSCF:----------  Subscription list end -----------
 5(5049) INF:P-CSCF:----------  Registrar Contents begin --------
 5(5049) INF:P-CSCF:----------  Registrar Contents end ----------
 5(5049) INF:P-CSCF:----------  Subscription list begin ---------
 5(5049) INF:P-CSCF:----------  Subscription list end -----------
 5(5049) INF:P-CSCF:----------  Registrar Contents begin --------
 5(5049) INF:P-CSCF:----------  Registrar Contents end ----------
 5(5049) INF:P-CSCF:----------  Subscription list begin ---------
 5(5049) INF:P-CSCF:----------  Subscription list end -----------
 5(5049) INF:P-CSCF:----------  P-CSCF Dialog List begin --------------
 5(5049) INF:P-CSCF:----------  P-CSCF Dialog List end   --------------
 5(5049) INF:P-CSCF:----------  Registrar Contents begin --------
 5(5049) INF:P-CSCF:----------  Registrar Contents end ----------
 5(5049) INF:P-CSCF:----------  Subscription list begin ---------
 5(5049) INF:P-CSCF:----------  Subscription list end -----------
 5(5049) INF:P-CSCF:----------  Registrar Contents begin --------
 5(5049) INF:P-CSCF:----------  Registrar Contents end ----------
 5(5049) INF:P-CSCF:----------  Subscription list begin ---------
 5(5049) INF:P-CSCF:----------  Subscription list end -----------
 5(5049) INF:P-CSCF:----------  Registrar Contents begin --------
 5(5049) INF:P-CSCF:----------  Registrar Contents end ----------
 5(5049) INF:P-CSCF:----------  Subscription list begin ---------
 5(5049) INF:P-CSCF:----------  Subscription list end -----------
```

(**c**)

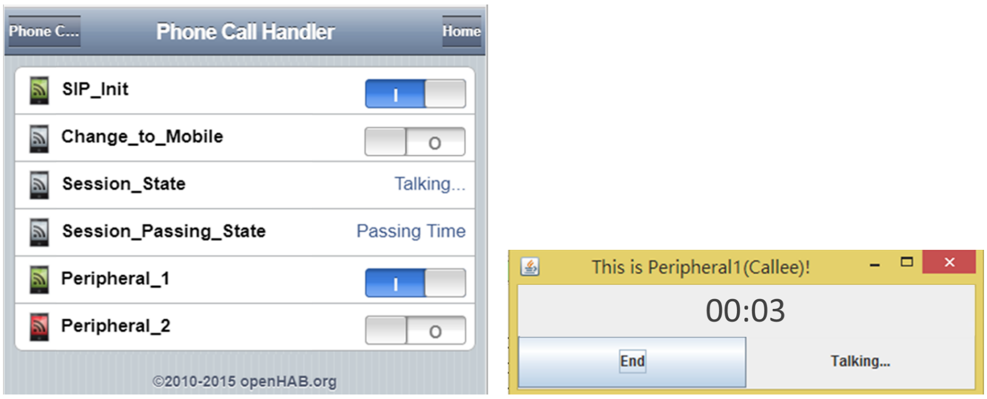
jinghong@jinghong-OptiPlex-380: /opt/OpenIMSCore

```
 5(5126) INF:S-CSCF:----------  Registrar Contents begin --------
 5(5126) INF:S-CSCF:----------  Registrar Contents end ----------
14(5135) --- Peer List: ---
14(5135)  S[I_Open] hss.open-ims.test:3868 D[ ]
14(5135)         [16777216,10415]
14(5135)         [16777216,4491]
14(5135)         [16777216,13019]
14(5135)         [16777216,0]
14(5135)         [16777217,10415]
14(5135)         [16777221,10415]
14(5135) ------------------
 5(5126) INF:S-CSCF:----------  Registrar Contents begin --------
 5(5126) INF:S-CSCF:----------  Registrar Contents end ----------
14(5135) DBG:peer_timer(): Peer hss.open-ims.test       State 6
14(5135) --- Peer List: ---
14(5135)  S[I_Open] hss.open-ims.test:3868 D[ ]
14(5135)         [16777216,10415]
14(5135)         [16777216,4491]
14(5135)         [16777216,13019]
14(5135)         [16777216,0]
14(5135)         [16777217,10415]
14(5135)         [16777221,10415]
14(5135) ------------------
```

(**d**)

Figure 9. Practical implementation of each IMS unit: (**a**) HSS; (**b**) I-CSCF; (**c**) P-CSCF, and (**d**) S-CSCF.

For a personal self IMS device, the functions for control of OSGi such as (1) IMS service proxy initialization (showed by SIP_Init in Figure 10); (2) IMS session available to a peripheral device (showed by Peripheral_1 in Figure 10); and (3) change to a direct mode (showed by Change_to_Mobile in Figure 10) as well as other functions of tracking "IMS session state (shown by Session_State in Figure 10)" and/or "peripheral device's session passing state (shown by Session_Passing_State in Figure 10)" are shown on the user interface of the user equipment, for example, a smart device, as shown in Figure 10. A session request is made by a user from a peripheral device through a sensor or a touch user interface, which is implemented in the experimental environment, as shown in Figure 10b.



SIP_Init I

Change_to_Mobile O

Session_State Talking...

Session_Passing_State Passing Time

Peripheral_1 I

Peripheral_2 O

(**a**)

This is Peripheral1(Callee)!

00:03

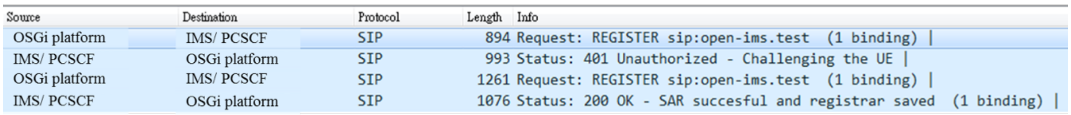
Talking...

(**b**)

Figure 10. Application program for (**a**) the personal self IMS to session changes on the IMS proxy platform; and (**b**) peripheral device.

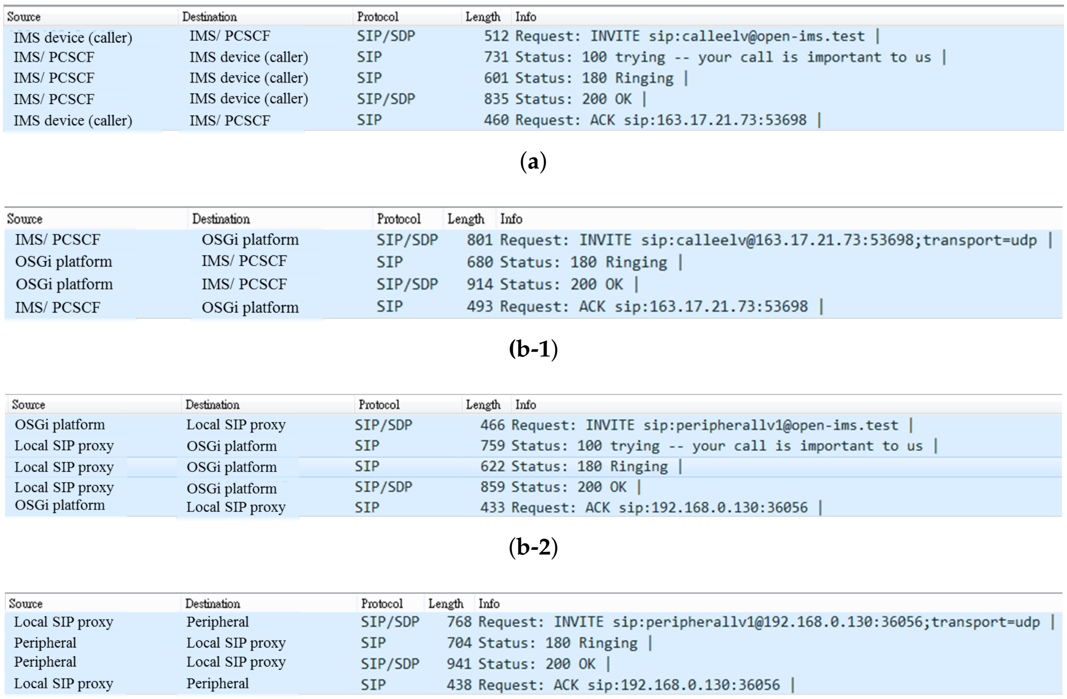
In this paper, the "Wireshake" is used to capture network packets with which the running services are displayed.

- Registration for services on the IMS service proxy platform: The service proxy of a user's personal self IMS device is acquired by the IMS service proxy gateway through IMS/IMPU sharing. The captured packets by Wireshake depicted in Figure 11.
- A phone session between a peripheral device in the field domain and other users (proxy mode) is enabled through steps as follows: (a) a session invitation initiated by a Callee; (b) the session invitation accepted by the IMS service proxy platform for session passing between a peripheral device and the IMS service proxy platform; and (c) the peripheral device responding to the session passing for exchanges of media resources with the Callee. The captured packets by Wireshake depicted in Figrue 12.



Source	Destination	Protocol	Length	Info
OSGi platform	IMS/ PCSCF	SIP	894	Request: REGISTER sip:open-ims.test (1 binding) |
IMS/ PCSCF	OSGi platform	SIP	993	Status: 401 Unauthorized - Challenging the UE |
OSGi platform	IMS/ PCSCF	SIP	1261	Request: REGISTER sip:open-ims.test (1 binding) |
IMS/ PCSCF	OSGi platform	SIP	1076	Status: 200 OK - SAR succesful and registrar saved (1 binding) |

Figure 11. Registration for services on the IMS service proxy platform.



Source	Destination	Protocol	Length	Info
IMS device (caller)	IMS/ PCSCF	SIP/SDP	512	Request: INVITE sip:calleelv@open-ims.test |
IMS/ PCSCF	IMS device (caller)	SIP	731	Status: 100 trying -- your call is important to us |
IMS/ PCSCF	IMS device (caller)	SIP	601	Status: 180 Ringing |
IMS/ PCSCF	IMS device (caller)	SIP/SDP	835	Status: 200 OK |
IMS device (caller)	IMS/ PCSCF	SIP	460	Request: ACK sip:163.17.21.73:53698 |

(**a**)

Source	Destination	Protocol	Length	Info
IMS/ PCSCF	OSGi platform	SIP/SDP	801	Request: INVITE sip:calleelv@163.17.21.73:53698;transport=udp |
OSGi platform	IMS/ PCSCF	SIP	680	Status: 180 Ringing |
OSGi platform	IMS/ PCSCF	SIP/SDP	914	Status: 200 OK |
IMS/ PCSCF	OSGi platform	SIP	493	Request: ACK sip:163.17.21.73:53698 |

(**b-1**)

Source	Destination	Protocol	Length	Info
OSGi platform	Local SIP proxy	SIP/SDP	466	Request: INVITE sip:peripherallv1@open-ims.test |
Local SIP proxy	OSGi platform	SIP	759	Status: 100 trying -- your call is important to us |
Local SIP proxy	OSGi platform	SIP	622	Status: 180 Ringing |
Local SIP proxy	OSGi platform	SIP/SDP	859	Status: 200 OK |
OSGi platform	Local SIP proxy	SIP	433	Request: ACK sip:192.168.0.130:36056 |

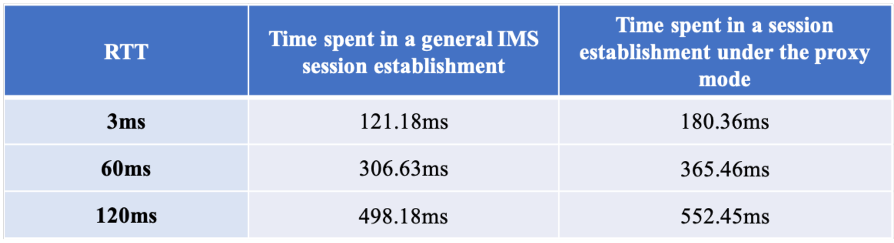
(**b-2**)

Source	Destination	Protocol	Length	Info
Local SIP proxy	Peripheral	SIP/SDP	768	Request: INVITE sip:peripherallv1@192.168.0.130:36056;transport=udp |
Peripheral	Local SIP proxy	SIP	704	Status: 180 Ringing |
Peripheral	Local SIP proxy	SIP/SDP	941	Status: 200 OK |
Local SIP proxy	Peripheral	SIP	438	Request: ACK sip:192.168.0.130:36056 |

(**c**)

Figure 12. A voice call session procedure: (**a**) a session invitation initiated by a Callee; (**b**) the session invitation accepted by the IMS service proxy platform for session passing between a peripheral device and the IMS service proxy platform; and (**c**) the peripheral device responding to the session passing for exchanges of media resources with the Callee.

With Round-trip time (RTT) of 3 ms, 60 ms and 120 ms configured, the time consumed in a general IMS session establishment and the time consumed in a session establishment under the proxy mode are measured for displays of sessions between two parties in three different network environments, for example, the two parties in the same domain (RTT = 3 ms) or in different domains and at different distances (RTT = 60 ms or 120 ms). Figure 13 illustrates the time consumed in setup of an IMS session and the time consumed in setup of a session under the proxy mode for three types of RTT. Compared with the time consumed in setup of a general IMS session, the additional 60 ms is consumed in a session establishment under the proxy mode for the step of Session passing specified in Section 4.2.



RTT	Time spent in a general IMS session establishment	Time spent in a session establishment under the proxy mode
3ms	121.18ms	180.36ms
60ms	306.63ms	365.46ms
120ms	498.18ms	552.45ms

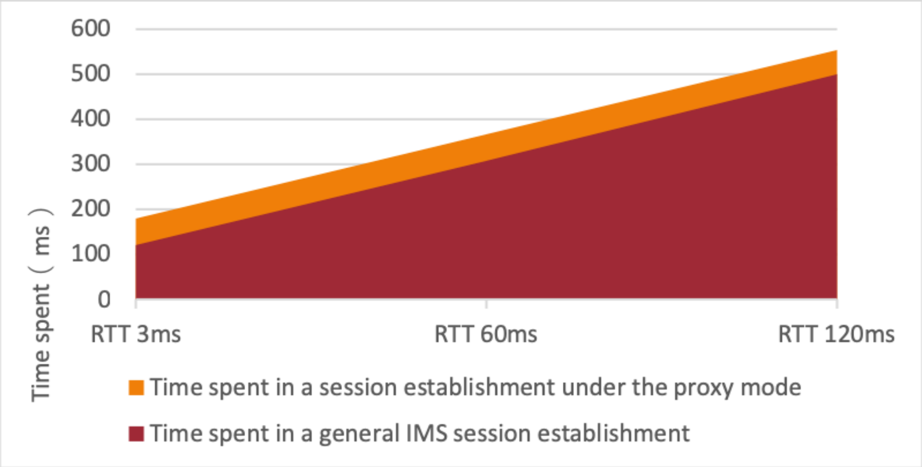


Figure 13. Comparison of time consumed in setup of a general IMS session and time consumed in setup of a session under the proxy mode.

6. Conclusions

In this paper, we realize the IMS service proxy gateway for non-IMS devices accessing IMS services in the trusted domain, such as home or office. The software and hardware services based on OSGi as the intermediate platform are integrated in this paper for effectuating the IMS service proxy gateway, which supports IMS services by implementation of the IMS Bundle and the RTP Bundle, fulfills IMS services shared by different devices on the basis of IMPU Sharing and grants the privilege to the IMS service proxy gateway. In addition, we adopt Caller preference and Q-value to no multiple responses attributed to SIP Forking. Because of more peripheral devices with distinctive conditions and capacities in the IoT era, a gateway controlling these peripheral devices' conditions for fast and effective applications is the key point in future studies.

Author Contributions: Conceptualization, Y.-C.C and J.-W.L.; Software, J.-H.L.; Supervision, Writing - original draft, J.-W.L.; Writing - review & editing, Y.-C.C and J.-W.L.

Funding: The research was supported by the Ministry of Science and Technology of the Republic of China under Grant No. MOST 107-3011-F-241 -001 and MOST 106-2221-E-324 -004.

Conflicts of Interest: The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

Abbreviations are arranged in alphabetical order.

3GPP	The 3rd Generation Partnership Project
AS	Application Server
CSCF	Call Session Control Function
DLNA	Digital Living Network Alliance
HSS	Home Subscriber Server
I-CSCF	Interrogating-Call Session Control Function
ICT	Information and Communication Technology
IMS	IP Multimedia Subsystem
IoT	Internet on Things
IMPI	IMS Private User Identity
IMPU	IMS Public User Identity
JMF	Java Media Framework
JADE	Java Agent DEvelopment
NGN	Next Generation Network
OSGI	Open Service Gateway Initiative
P2P	Peer-to-Peer
P-CSCF	Proxy-Call Session Control Function
RTT	Round-Trip Time
RTP	Real-Time Transport Protocol
REST	Representational State Transfer
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
S-CSCF	Serving-Call Session Control Function
VoIP	Voice over IP
USIM	Universal Subscriber Identity Module
UI	User Interface
UE	User Equipment
XML	Extensible Markup Language

References

1. Camarillo, G.; Garcia-Martin, M.A. Wiley InterScience (Online Service). *The 3G IP Multimedia Subsystem (IMS): Merging the Internet And the Cellular Worlds*; J. Wiley & Sons: Hoboken, NJ, USA, 2008; p. 618.
2. Olsson, U.; Mulligan, C.; Fikouras, I.; Ryde, A.; Stille, M.; Noldus, R. *IMS Application Developer's Handbook*; Academic Press: Cambridge, MA, USA, 2011.
3. Copeland, R. *Converging NGN Wireline and Mobile 3G Networks with IMS*; Auerbach Publications: New York, NY, USA, 2008; p. 484.
4. Ahson, S.; Ilyas, M. *IP Multimedia Subsystem (IMS) Handbook*; CRC Press: Boca Raton, FL, USA, 2009.
5. Li, J.W.; Jan, Y.S.; Chang, Y.C. Multimedia synchronization on IP multimedia subsystem to support collaborative learning. In Proceedings of the 2016 5th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2016, Kumamoto, Japan, 10–14 July 2016; pp. 1182–1183
6. OSGi. Open Service Gateway Initiative (OSGi): The Dynamic Module System for Java. Available online: https://www.osgi.org/ (accessed on 22 October 2018).
7. Chang, C.Y.; Kuo, C.H.; Chen, J.C.; Wang, T.C. Design and Implementation of an IoT Access Point for Smart Home. *Appl. Sci.* **2015**, 5, 1882–1903, doi:10.3390/app5041882. [CrossRef]
8. Cheng, B.; Wei, Z. Restful m2m gateway for remote wireless monitoring for district central heating networks. *Sensors* **2014**, 22447–22470, doi:10.3390/s141222447. [CrossRef] [PubMed]
9. Sebbak, F.; Benhammadi, F. Majority-consensus fusion approach for elderly IoT-based healthcare applications. *Ann. Telecommun.* **2017**, *72*, 157–171, doi:10.1007/s12243-016-0550-7. [CrossRef]
10. Hosek, J.; Masek, P.; Kovac, D.; Ries, M.; Kröpfl, F. IP home gateway as universal multi-purpose enabler for smart home services. *Elektrotech. Inf.* **2014**, *131*, 123–128, doi:10.1007/s00502-014-0209-x. [CrossRef]

11. Bhatti, J.; Van Den Wouwer, D.; Kerckhove, W.; Dupont, T.; Volckaert, B. A scalable software framework for real-Time data processing in the railway environment. In Proceedings of the 2016 IEEE International Conference on Intelligent Rail Transportation, ICIRT 2016, Birmingham, UK, 23–25 August 2016; pp. 170–176.
12. Jeong, Y.; Son, S.; Jeong, E.; Lee, B.; Jeong, Y.; Son, S.; Jeong, E.; Lee, B. An Integrated Self-Diagnosis System for an Autonomous Vehicle Based on an IoT Gateway and Deep Learning. *Appl. Sci.* **2018**, *8*, 1164, doi:10.3390/app8071164. [CrossRef]
13. Lee, C.T.; Shen, T.C.; Lee, W.D.; Lee, C.T.; Shen, T.C.; Lee, W.D. A Novel Optical Morse Code-Based Electronic Lock Using the Ambient Light Sensor and Fuzzy Controller. *Appl. Sci.* **2017**, *7*, 140, doi:10.3390/app7020140. [CrossRef]
14. Lai, C.F.; Chen, M.; Vasilakos, A.V.; Huang, Y.M. Extending the DLNA-based multimedia sharing system to P2P network on OSGi frameworks. In Proceedings of the GLOBECOM—IEEE Global Telecommunications Conference, Miami, FL, USA, 6–10 December 2010; pp. 1–5.
15. Lai, C.F.; Huang, Y.M.; Chao, H.C. DLNA-based multimedia sharing system for OSGI framework with extension to P2P network. *IEEE Syst. J.* **2010**, *4*, 262–270, doi:10.1109/JSYST.2010.2047175. [CrossRef]
16. Andrade, J.C.; Ribeiro, R.B.; Villaca, R.S.; Martinello, M.; Santos, C.A. SANGN: A new service oriented architecture for provisioning of NGN scalable multimedia services. In Proceedings of the International Conference on Advanced Information Networking and Applications, AINA, Matsue, Japan, 16–18 May 2018; pp. 504–511.
17. Cabelino Ribeiro, R.B.; Martinello, M.; Santos, C.A.S.; Soares, R.B. An architectural framework for delivering SIP-AS multimedia services based on JADE/OSGi technology. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*; Springer: Berlin, Germany, 2015; pp. 122–128.
18. Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; Johnston, A.; Peterson, J.; Sparks, R.; Handley, M.; Schooler, E. RFC 3261: SIP: Session Initiation Protocol. 2002. doi:10.17487/rfc3261. Available online: http://www.rfc-editor.org/info/rfc3261 (accessed on 22 October 2018).
19. Fielding, R.T. Architectural Styles and the Design of Network-based Software Architectures. *Building* **2000**, *54*, 162. [CrossRef]
20. Rosenberg, J.; Schulzrinne, H.; Kyzivat, P. RFC 3840: Indicating User Agent Capabilities in the Session Initiation Protocol (SIP). 2004. Available online: https://datatracker.ietf.org/doc/rfc3840/ (accessed on 22 October 2018).
21. Rosenberg, J.; Schulzrinne, H.; Kyzivat, P. RFC 3841: Caller Preferences for the Session Initiation Protocol (SIP). 2004. Available online: https://www.rfc-editor.org/info/rfc3841 (accessed on 22 October 2018).
22. Rosenberg, J. RFC 5688: A Session Initiation Protocol (SIP) Media Feature Tag for MIME Application Subtypes. 2010. Available online: https://www.rfc-editor.org/info/rfc5688 (accessed on 22 October 2018).
23. Equinox, E. Eclipse Equinox-OSGi Core Framework. Available online: http://www.eclipse.org/equinox/ (accessed on 22 October 2018).
24. O'Doherty, P.; Ranganathan, M. *JAIN SIP Tutorial: Serving the Developer Community*; Sun Microsystems: Santa Clara, CA, USA, 2003.
25. Simpson-Young, B. Programming with the Java media framework. *Int. Res.* **1999**, *9*, 408–418, doi:10.1108/intr.1999.17209baf.004. [CrossRef]